CMPSC 465 — Data Structures & Algorithms
Spring 2024 — Mehrdad Mahdavi and David Koslicki

Worksheet 2

Wednesday, January 24, 2023

1. **Growth Rate.** Sort the following expressions from slowest to fastest growth rate. (You may assume all logarithms have base 2.)

(a) $(\sqrt{2})^{\log n}$
(b) n^2
(c) $n!$
(d) $(\log n)!$
(e) $(\frac{3}{2})^n$
(f) $(\log n)^2$
(g) n^3
(h) $\log(n!)$
(i) 2^{2^n}
(j) $n^{\frac{1}{\log n}}$
(k) $\log\log n$
(l) $n2^n$
(m) $n^{(\log\log n)^2}$
(n) $2^{2^{2n+1}}$
(o) $2^{\log n}$
(p) $2^{\sqrt{2\log n}}$
(q) $\sqrt{\log n}$

2. **Find run time.** How long does the recursive multiplication algorithm (given below) take to multiply a non-negative n-bit number by a non-negative m-bit number? Justify your answer.

Algorithm 1 multiply(x,y)

Input: An n-bit integer x and an m-bit integer y, where $x, y \geq 0$
Output: Their product $x \cdot y$
if $y = 0$ **then**
  return 0
end if
z = multiply(x, $\lfloor \frac{y}{2} \rfloor$)
if y = even **then**
  return $2z$
else
  return $x + 2z$
end if

3. **Computing Factorials.** Consider the problem of computing $N! = 1 \times 2 \times \cdots \times N$.

   1. If N is a b-bit number, how many bits long is $N!$ (Θ notation suffices)?
   2. Consider the simple algorithm to compute $N!$ that does the multiplication in sequence, $1 \times 2 \times 3 \times \ldots \times N$. Analyze its running time.

4. **Fibonacci growth.** The Fibonacci numbers $F_0, F_1, F_2 \ldots$ are defined by the recurrence $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$. In this problem, we will confirm that this sequence grows exponentially fast and obtain some bounds on its growth.

(a) Use induction to prove that $F_n \geq 2^{0.5n}$ for $n \geq 6$.

(b) Find a constant $c > 0$ such that $F_n \geq 2^{cn}$ for all $n \geq 3$. Show that your answer is correct.

(c) Find the maximum constant $c > 0$ for which $F_n = \Omega(2^{cn})$.